;====================================================================

;

; Author : ADI - Apps

;

; Date : 2nd July 2003

;

; File : TempUart.asm

;

; Hardware : ADuC845

;

; Description : This Program takes a temperature measurement every

; second from the on-chip temp sensor and sends the

; temp in degrees Celcius up the UART to the PC where

; it can be read using hyperterminal

;====================================================================

;

$mod845 ; Use 8052 / ADuC834 predefined Symbols

LED EQU P3.4

FLAG EQU 00h

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; DEFINE VARIABLES IN INTERNAL RAM

DSEG

ORG 0033h

COUNT1: DS 1

COUNT2: DS 1

COUNT3: DS 1

DIG1: DS 1

DIG2: DS 1

DIG3: DS 1

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; BEGINNING OF CODE

CSEG

ORG 0000H

JMP MAIN

;====================================================================

ORG 0060H ; Start code at address above interrupts

MAIN: ; Main program

MOV T3CON,#83h

MOV T3FD,#12h

MOV SCON,#52h

; Configure ADC

MOV ADCMODE, #10H ; ENABLE AUX Mode - Power down

MOV ADC1CON, #0EH ; USE INTERNAL REFERENCE

; PTAT(+) --> PTAT(-)

; BIPOLAR MODE on Temp Sensor

; Fixed +/- 1.25V range

MOV ADCMODE, #14h ;Aux ADC enabled, OFS cal

MOV ADCMODE, #15h ;Aux ADC enabled, GN cal

MOV DPTR, #TITLE

CALL SENDSTRING ; write title block on screen

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; TEMP MEASURE LOOP

TEMPLOOP:

MOV ADCMODE, #12H ; INITIATE A SINGLE AUX CONV

JNB RDY1,$ ; Wait for conversion results

; conversion result ready

; a value of 80h in AD1H=0degC

MOV A, ADC1H ; 80h=0, FFh=+127, 00h=-128

CLR C

SUBB A, #80H ; convert to 2's comp

; FFh=-1, 80h=-128, 00h=0, 7Fh=+127

SENDDECs: ; SENDs the signed decimal number in Acc up UART

; -128->127

PUSH B

PUSH ACC

JNB ACC.7, HUNDREDS

MOV A, #'-' ; transmit minus sign

CALL SENDCHAR

POP ACC ; restore original value of A

PUSH ACC ; remember original value of A

CPL A

INC A

HUNDREDS: ; check #hundreds

MOV B, #100 ; divide remainder by 100

DIV AB ; A receives integer quotient

; B receives the remainder

SETB F0

JZ TENS ; if ACC=0 then num=0xx

CLR F0

ADD A, #'0'

LCALL SENDCHAR

TENS: ; check tens

MOV A,B

MOV B,#10

DIV AB ; divide remainder by 10

JNB F0, SEND0 ; if F0 is cleared the a number

; exists in the 100s

JZ UNITS

SEND0: ADD A, #'0' ; only send a zero if number

CALL SENDCHAR ; existed in the 100s

UNITS: MOV A,B ; send remainder (even if 0)

ADD A, #'0'

CALL SENDCHAR

POP ACC

POP B

MOV DPTR, #DEGREES

CALL SENDSTRING

MOV A, #07 ;Give 0.503s delay

CALL DELAY

JMP TEMPLOOP

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; DELAY

DELAY: ; Delays by 71.28ms \* A

; 83mSec based on 1.57MHZ Core Clock

MOV R2,A ; Acc holds delay variable

DLY0: MOV R3,#220 ; Set up delay loop0

DLY1: MOV R4,#255 ; Set up delay loop1

DJNZ R4,$ ; Dec R4 & Jump here until R4 is 0

; wait here for 255\*1.27us=324us

DJNZ R3,DLY1 ; Dec R3 & Jump DLY1 until R3 is 0

; Wait for 220\*324us = 71.28ms

DJNZ R2,DLY0 ; Dec R2 & Jump DLY0 until R2 is 0

; wait for ACC\*71.28ms

RET ; Return from subroutine

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; SENDCHAR

SENDCHAR: ; sends ASCII value contained in A to UART

JNB TI,$ ; wait til present char gone

CLR TI ; must clear TI

MOV SBUF,A

RET

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; SENDSTRING

SENDSTRING: ; sends ASCII string to UART starting at location

; DPTR and ending with a null (0) value

PUSH ACC

PUSH B

CLR A

MOV B,A

IO0010: MOV A,B

INC B

MOVC A,@A+DPTR

JZ IO0020

CALL SENDCHAR

JMP IO0010

IO0020: POP B

POP ACC

RET

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DEGREES: DB ' degrees C',10,13,0

TITLE: DB 10,10,13,'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_',10,13

DB 'Analog Devices MicroConverter ADuC845',10,13

DB ' Temp Sensor Demo Routine',10,13,0

END